REMARKS

Claims 1-32 were examined in a subject Office action dated 04-May-2007. Claims 1, 6, 8, 9, 11, 12 14, 16, and 22-25 have been amended, claims 7, 10, 26 and 27 have been canceled, claims 33-34 have been added, and claims 2-5, 13, 15, 17-21 and 28-32 are currently pending in the subject application and are presently under consideration as shown on pp. 2-6 of the Reply. Applicant's representative appreciates the opportunity to discuss the pending Office action afforded on 27 June 2007. The amendments and remarks contained herein encompass the remarks made at that time.

Favorable reconsideration of the subject patent application is respectfully requested in view of the comments and amendments herein.

I. Rejection of Claims 1-12 Under 35 U.S.C. §112

Claims 1-12 stand rejected under 35 U.S.C. §112 as being incomplete for omitting essential structural cooperative relationships of elements, such omission amounting to a gap between the necessary structural connections. Claim 1 has been amended to clarify how each components operates upon a user input expression to show the cooperation there between to obviate the rejection. Claim 6 has been amended to delete computer code to obviate the rejection under §112 as being indefinite. Withdrawal of the rejection under §112 is respectfully requested.

II. Rejection of Claims 1-6, 8-9, and 11-25 Under 35 U.S.C. §102(b)

Claims 1-6, 8-9, and 11-25 stand rejected under 35 U.S.C. §102(b) as being anticipated by Acton, *et al.* (6,209,040). It is respectfully submitted that this rejection should be withdrawn for at least the following reasons. Acton does not teach or suggest each and every limitation of appellants' claimed invention.

> For a prior art reference to anticipate, 35 U.S.C. §102 requires that "each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." In re Robertson, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950 (Fed. Cir. 1999) (quoting Verdegaal Bros., Inc. v. Union Oil Co., 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987)).

As a general overview, aspects of the present Application are directed in part to allowing a user to define a type or class as part of a generalized comprehension that includes a collection and an operation that is performed upon the collection. Thereby, users are able to more intuitively program via a method that bridges between the two worlds of data integration (e.g., relational tables and semi-structured XML trees) and fixed programming structures of imperative, object-oriented programming languages. (See paras. [0009]-[0010]). By virtue of this ability to input source code in a generalized comprehension, the following advantages and others are realized: (a) the introduction of comprehension notation in an imperative, object-oriented language, (b) the ability for users to overload the comprehension notation by implementing certain interfaces or patterns, (c) the ability to special case the translation of the comprehension notation for particular source and result types (d) for alternative syntaxes that have similar functional patterns, including select, the ability to omit type declarations, and so forth. (See para. [0027].)

By contrast, Acton is directed to storing type information in a language-independent manner (see col. 1, lines 7-10) to address the problem of allowing a second computer program to access the objects instantiated in shared memory by a first program (see Acton col. 1, lines 38-50). The interface taught by Acton is a type library provided by the first program that is compiled with the second program so that when the compiled second program is executed, it accesses the type information of the first program (see col. 2, lines 37-48).

Turning to independent claim 1, the claim as amended recites a system having a language component that receives a user input expression comprising comprehension notations, enabling programming of comprehension notations in the imperative language. An interface component describes a meaning of the comprehension notations. A translation component analyzes the meaning of the comprehension notations and facilitates execution of the comprehension notations in accordance with the imperative language. Thereby, a programming environment is provided providing the power of generalized comprehension while taking advantage of a programming infrastructure of an imperative language.

In rejecting claim 1 as anticipated by Acton, the Examiner relied upon the

Abstract of Acton:

> A method and system for interfacing to type libraries are provided. In a preferred embodiment, the present invention defines an interface to a type library. The interface has a plurality of interface methods through which type information can be stored in and retrieved from the type library. A plurality of implementations of the defined interface are provided. A computer program is compiled using the defined interfaces to access the type information. When the compiled program is executed, it accesses the type information using one of the plurality of implementations. In a preferred embodiment, a type library contains type information to allow a compiler to bind to an instance of type at compile type (early binding).

Acton thus fails to address allowing a user to input comprehension notations but rather allows a user to access a pre-defined type library to interface to a second computer program. In addition to failing to teach or suggest each limitation of the claimed invention, Acton is directed to solving a different problem. Thus, the cited references fail to provide a suggestion or motivation to make the modification necessary to Acton to realize the claimed invention, and thus would be patentable over Acton. Reconsideration and allowance of claim 1 is requested, as well as for claims 2-6, 8, 9, 11-13, and 33-35 that depend there from.

Turning to independent claim 14, the claim as amended recites in part a means for receiving a user input generalized comprehension expression, the generalized comprehension expression defined exterior to the list comprehension. In rejecting claim 14, the Examiner looked to Acton at col. 2, lines 44-48, which pertains to early binding to an instance of type at compile time. However, the means as claimed for receiving a user input generalized comprehension expression not suggested or taught by Acton. Reconsideration and allowance of claim 14 is respectfully requested, as well as for claim 15 that depends there from.

Turning to independent claim 16, the claim as amended recites a method for providing a programming environment for a user to input a generalized comprehension that is automatically translated into a language form in an imperative language. This is accomplished by defining a list comprehension expression and defining a generalized comprehension class as an exterior component to the list comprehension expression within an imperative language environment. Rejection of claim 16 was given as being the same as for claim 1, and thus for the reasons given above, reconsideration and allowance of claim 16 is respectfully requested, as well as for claims 17-21 that depend there from.

Turning to independent claim 22, the claim as amended recites a medium on which a data structure is stored comprising a first data field that defines a static comprehension notation in the imperative language. A second data field defines a generalized comprehension notation received from a user input. A third data field links the static comprehension notation with the generalized comprehension notation. The claim has been further clarified by stating that the second data field comprising the static comprehension notation and a comprehension notation external to the static comprehension notation to form a generalized comprehension.

In rejecting claim 22, the Examiner found these features to be inherent in an abstract class and rejected claim 22 on the same basis as claim 3, which was rejected in turn with the comment: "See that the interface is an abstract class (generalized)." However, it is noted that an abstract type is a type in a nominative type system which is declared by the programmer, and which has the property that it contains no members which are also not members of some declared subtype. However, the claim as amended differentiates an abstract class from the generalized comprehension and thus is not anticipated by Acton. Reconsideration and allowance of claim 22 is respectfully requested,

III. <u>Rejection of Claims 7, 10 and 26-32 Under 35 U.S.C. §103(a)</u>

Claims 7, 10 and 26-32 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Acton in view of the applicant's choice of selection of a specific program, with its inherent functions, to develop his interface. In response thereto, claims 7, 10, 26 and 27. Claims 28-32 depend from base claim 22 and should be allowed for the reasons given above for claim 22.

CONCLUSION

The present application is believed to be in condition for allowance in view of the above comments and amendments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063.

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicants' undersigned representative at the telephone number below.

Respectfully submitted,
AMIN, TUROCY & CALVIN, LLP

/Himanshu S. Amin/
Himanshu S. Amin
Reg. No. 40,894

AMIN, TUROCY & CALVIN, LLP
24TH Floor, National City Center
1900 E. 9TH Street
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731